

Package ‘tipr’

September 10, 2021

Type Package

Title Tipping Point Analyses

Version 0.3.0

Author Lucy D'Agostino McGowan

Maintainer Lucy D'Agostino McGowan <lucydagostino@gmail.com>

Description The strength of evidence provided by epidemiological and observational studies is inherently limited by the potential for unmeasured confounding.

We focus on three key quantities: the observed bound of the confidence interval closest to the null, a plausible residual effect size for an unmeasured continuous or binary confounder, and a realistic mean difference or prevalence difference for this hypothetical confounder. Building on the methods put forth by Lin, Psaty, & Kronmal (1998) <[doi:10.2307/2533848](https://doi.org/10.2307/2533848)>, we can use these quantities to assess how an unmeasured confounder may tip our result to insignificance, rendering the study inconclusive.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Suggests testthat, broom, dplyr, MASS

Imports glue, tibble, purrr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-09-10 08:00:02 UTC

R topics documented:

lm_tip	2
observed_bias_order	3
observed_bias_tbl	4
observed_bias_tip	5
observed_covariate_e_value	6
tip	6
tipr	8
tip_with_binary	8

lm_tip *Tip a linear model result with a continuous confounder.*

Description

choose one of the following, and the other will be estimated:

- `smd`
- `outcome_association`

Usage

```
lm_tip(
  d,
  smd = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)
```

Arguments

<code>d</code>	Data frame. A data frame with the observed effect(s). This should have at least two columns with the lower and upper confidence bounds. These columns are assumed to be called <code>conf.low</code> and <code>conf.high</code> . If this is not the case, the names can be changed using the <code>lb_name</code> and <code>ub_name</code> parameters.
<code>smd</code>	Numeric. Estimated scaled mean difference between the unmeasured confounder in the exposed population and unexposed population
<code>outcome_association</code>	Numeric positive value. Estimated association between the unmeasured confounder and the outcome
<code>verbose</code>	Logical. Indicates whether to print informative message. Default: TRUE
<code>lb_name</code>	Character. Column name of <code>d</code> that holds the lower confidence bound. Default: <code>conf.low</code> based on broom defaults.
<code>ub_name</code>	Character. Column name of <code>d</code> that holds the upper confidence bound. Default: <code>conf.high</code> based on broom defaults.

Value

Data frame.

Examples

```
d <- data.frame(conf.low = 1.2, conf.high = 1.5)
## to estimate the association between an unmeasured confounder and outcome
## needed to tip analysis
lm_tip(d, smd = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
lm_tip(d, smd = -2, outcome_association = -0.05)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  lm(wt ~ mpg, data = mtcars) %>%
  broom::tidy(conf.int = TRUE) %>%
  dplyr::filter(term == "mpg") %>%
  lm_tip(outcome_association = 2.5)
}
```

observed_bias_order *Order observed bias data frame for plotting*

Description

Order observed bias data frame for plotting

Usage

```
observed_bias_order(d, by)
```

Arguments

d Observed bias data frame. Must have columns dropped and type

by Character. Variable in d to order by.

Value

Data frame in the correct order

observed_bias_tbl *Create a data frame to assist with creating an observed bias plot*

Description

Create a data frame to assist with creating an observed bias plot

Usage

```
observed_bias_tbl(ps_mod, outcome_mod, drop_list = NULL)
```

Arguments

ps_mod	Model object for the propensity score model
outcome_mod	Model object for the outcome model
drop_list	Named list of covariates or groups of covariates to drop if NULL, will default to dropping each covariate one at a time.

Value

Data frame with the following columns:

- dropped: The covariate or group of covariates that were dropped
- type: Explanation of dropped, whether it refers to a single covariate (covariate) or a group of covariates (group)
- ps_formula: The new formula for the updated propensity score model
- outcome_formula: The new formula for the updated outcome model
- ps_model: The new model object for the updated propensity score model
- p: The updated propensity score

Examples

```
ps_mod <- glm(am ~ mpg + cyl + I(hp^2), data = mtcars)
outcome_mod <- lm(qsec ~ am + hp + disp + wt, data = mtcars)
observed_bias_tbl(
  ps_mod,
  outcome_mod,
  drop_list = list(
    group_one = c("mpg", "hp"),
    group_two = c("cyl", "wt")
  )
)
```

observed_bias_tip	<i>Create a data frame to combine with an observed bias data frame demonstrating a hypothetical unmeasured confounder</i>
-------------------	---

Description

Create a data frame to combine with an observed bias data frame demonstrating a hypothetical unmeasured confounder

Usage

```
observed_bias_tip(  
  tip,  
  point_estimate,  
  lb,  
  ub,  
  tip_desc = "Hypothetical unmeasured confounder"  
)
```

Arguments

tip	Numeric. Value you would like to tip to.
point_estimate	Numeric. Result estimate from the full model.
lb	Numeric. Result lower bound from the full model.
ub	Numeric. Result upper bound from the full model.
tip_desc	Character. A description of the tipping point.

Value

A data frame with five columns:

- `dropped`: the input from `tip_desc`
- `type`: Explanation of `dropped`, here `tip` to clarify that this was calculated as a tipping point.
- `point_estimate`: the shifted point estimate
- `lb`: the shifted lower bound
- `ub`: the shifted upper bound

observed_covariate_e_value

Calculate the Observed Covariate E-value

Description

Calculate the Observed Covariate E-value

Usage

```
observed_covariate_e_value(lb, ub, lb_adj, ub_adj, transform = NULL)
```

Arguments

lb	Numeric. The lower bound of the full model
ub	Numeric. The upper bound of the full model
lb_adj	Numeric. The lower bound of the adjusted model
ub_adj	Numeric. The upper bound of the adjusted model
transform	Character. If your effect is an odds ratio or hazard ratio, this will perform the transformation suggested by VanderWeele and Ding. Allowed values are: <ul style="list-style-type: none"> • "OR" • "HR"

Value

The Observed Covariate E-value

tip

Tip a result with a continuous confounder.

Description

choose one of the following, and the other will be estimated:

- smd
- outcome_association

Usage

```
tip(
  d,
  smd = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)

tip_with_continuous(
  d,
  smd = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)

tip_c(
  d,
  smd = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)
```

Arguments

d	Data frame. A data frame with the observed effect(s). This should have at least two columns with the lower and upper confidence bounds. These columns are assumed to be called <code>conf.low</code> and <code>conf.high</code> . If this is not the case, the names can be changed using the <code>lb_name</code> and <code>ub_name</code> parameters.
smd	Numeric. Estimated scaled mean difference between the unmeasured confounder in the exposed population and unexposed population
outcome_association	Numeric positive value. Estimated association between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE
lb_name	Character. Column name of <code>d</code> that holds the lower confidence bound. Default: <code>conf.low</code> based on broom defaults.
ub_name	Character. Column name of <code>d</code> that holds the upper confidence bound. Default: <code>conf.high</code> based on broom defaults.

Value

Data frame.

Examples

```
d <- data.frame(conf.low = 1.2, conf.high = 1.5)
## to estimate the association between an unmeasured confounder and outcome
## needed to tip analysis
tip(d, smd = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip(d, smd = -2, outcome_association = .99)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  glm(am ~ mpg, data = mtcars, family = "binomial") %>%
  broom::tidy(conf.int = TRUE, exponentiate = TRUE) %>%
  dplyr::filter(term == "mpg") %>%
  tip(outcome_association = 2.5)
}
```

tipr

tipr

Description

The tipr package.

References

D'Agostino McGowan, L. (2018). Improving Modern Techniques of Causal Inference: Finite Sample Performance of ATM and ATO Doubly Robust Estimators, Variance Estimation for ATO Estimators, and Contextualized Tipping Point Sensitivity Analyses for Unmeasured Confounding. PhD thesis, Vanderbilt University.

VanderWeele, TJ, and Peng D (2017). Sensitivity Analysis in Observational Research: Introducing the E-Value. *Ann Intern Med*, 167(4), 268–74.

Lin, DY, Psaty, BM, & Kronmal, RA. (1998). Assessing the sensitivity of regression results to unmeasured confounders in observational studies. *Biometrics*, 54(3), 948–963.

tip_with_binary

Tip a result with a binary confounder.

Description

Choose two of the following three to specify, and the third will be estimated:

- exposed_p
- unexposed_p
- outcome_association

Alternatively, specify all three and the function will return the number of unmeasured confounders specified needed to tip the analysis.

Usage

```
tip_with_binary(
  d,
  exposed_p = NULL,
  unexposed_p = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)
```

```
tip_b(
  d,
  exposed_p = NULL,
  unexposed_p = NULL,
  outcome_association = NULL,
  verbose = TRUE,
  lb_name = "conf.low",
  ub_name = "conf.high"
)
```

Arguments

d	Data frame. A data frame with the observed effect(s). This should have at least two columns with the lower and upper confidence bounds. These columns are assumed to be called <code>conf.low</code> and <code>conf.high</code> . If this is not the case, the names can be changed using the <code>lb_name</code> and <code>ub_name</code> parameters.
exposed_p	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population
unexposed_p	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population
outcome_association	Numeric positive value. estimated association between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE

lb_name	Character. Column name of d that holds the lower confidence bound. Default: conf.low based on broom defaults.
ub_name	Character. Column name of d that holds the upper confidence bound. Default: conf.high based on broom defaults.

Details

`tip_b()` is an alias for `tip_with_binary()`.

Examples

```
d <- data.frame(conf.low = 1.2, conf.high = 1.5)
## to estimate the association between an unmeasured confounder and outcome
## needed to tip analysis
tip_with_binary(d, exposed_p = 0.5, unexposed_p = 0)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip_with_binary(d, exposed_p = 0.5, unexposed_p = 0, outcome_association = 1.1)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  glm(am ~ mpg, data = mtcars, family = "binomial") %>%
  broom::tidy(conf.int = TRUE, exponentiate = TRUE) %>%
  dplyr::filter(term == "mpg") %>%
  tip_with_binary(exposed_p = 1, outcome_association = 1.15)
}
```

Index

`lm_tip`, [2](#)

`observed_bias_order`, [3](#)

`observed_bias_tbl`, [4](#)

`observed_bias_tip`, [5](#)

`observed_covariate_e_value`, [6](#)

`tip`, [6](#)

`tip_b(tip_with_binary)`, [8](#)

`tip_b()`, [10](#)

`tip_c(tip)`, [6](#)

`tip_with_binary`, [8](#)

`tip_with_binary()`, [10](#)

`tip_with_continuous(tip)`, [6](#)

`tipr`, [8](#)